Supplement for KmerStream

Páll Melsted*[1,2] Bjarni V. Halldórsson[2,3]
[1]Faculty of Industrial Engineering, Mechanical Engineering
and Computer Science, University of Iceland, Reykjavík, Iceland
[2]deCODE Genetics/Amgen, Reykjavík, Iceland
[3]School of Science and Engineering, Reykjavík University, Reykjavík, Iceland
pmelsted@hi.is

April 6, 2014

Theorem 1. *If $f_1 \geq \frac{F_0}{\lambda}$ then Algorithm 1 finds $\hat{f}_1$ such that $(1-\varepsilon)f_1 \leq \hat{f}_1 \leq (1+\varepsilon)f_1$ with probability at least $(1-\delta)$. The algorithm uses $O\left(\frac{\lambda^2 \log(1/\delta)}{\varepsilon^2}\log(N)\right)$ memory and $O(1)$ time per update, where N is the number of elements in the stream.*

The proof relies on the hash function being perfectly random, i.e. all hash values are independent random variables. In practice no such hash functions exist, but pairwise independent hash functions are close to being perfectly random when the underlying data has entropy [1]. If a hash function is δ'-close to being perfectly random, the difference in the probability of any event will be at most δ'. This allows us to analyze the probability as if the hash function were perfectly random, and simply add a δ' to the probability of failure at the end. In practice this difference, δ', is much smaller than the probability of failure.

Our data structure is a list T of $\log(n)$, arrays of length R, where $R = \frac{720\lambda^2\log(8/\delta)}{\varepsilon^2}$, we have not made any attempts to optimize the constant so as to not further complicate the proof. Each value in an array, T_w, is a 2-bit counter which can store the numbers from 0 to 3.

For each k-mer a we compute a hash value $h(a)$, where h is a hash function that gives 64-bit values and is guaranteed to give the same value for a and the reverse complement of a. For the value $h(a)$ we let w be the highest integer such that 2^{w-1} divides $h(a)$. Equivalently, w is the least-significant position that is 1 when $h(a)$ is written in binary. We let $j = \lfloor \frac{h(a)}{2^{w+1}} \rfloor$ and increment the value in $T_w[j]$ by one (if the value is already 3 we do nothing).

We can see that the value of w follows a geometric distribution $Geo(\frac{1}{2})$. Thus, half the k-mers will hash to the first array, a quarter to the second etc. To estimate f_1 we select the array, w^* that is closest to being half-full ($w^*\text{argmin}_w \left| |\{i : T_w[i] = 0\}| - \frac{1}{2} \right|$). Suppose N_{w^*} distinct k-mers hash to this array of R values. The probability that an element in the array has value 0 is then $\left(1 - \frac{1}{R}\right)^{N_{w^*}}$, suppose that x_1 of these N_{w^*} k-mers are singletons. The only way an element in the array has value 1 is if exactly one of the singletons hashed to this location. This occurs with probability

$$\frac{x_1}{R}\left(1-\frac{1}{R}\right)^{N_{w^*}-1} = \frac{x_1}{R-1}\left(1-\frac{1}{R}\right)^{N_{w^*}}$$

Let $\hat{p_0}$ and $\hat{p_1}$ be the fraction of elements with values 0 and 1 respectively. Then $\hat{p_0}$ and $\hat{p_1}$ are good estimates for the probability of a cell having value 0 or 1. We can estimate x_1 as

$$\hat{x_1} = (R-1)\frac{\hat{p_1}}{\hat{p_0}}$$

Since only a $\frac{1}{2^{w^*}}$ of all k-mers hashed to this particular array we can estimate f_1 as

$$\hat{f_1} = 2^{w^*}\hat{x_1} = 2^{w^*}(R-1)\frac{\hat{p_1}}{\hat{p_0}}$$

Given our choice of R, we need to show that

$$\begin{aligned}
&\frac{R}{4} \leq N_{w^*} \leq \frac{3R}{4}\\
&|p_0 - \hat{p_0}| \leq \frac{\varepsilon}{3}p_0\\
&|p_1 - \hat{p_1}| \leq \frac{\varepsilon}{3}p_1\\
&|2^{w^*}\hat{x_1} - f_1| \leq \frac{\varepsilon}{3}f_1
\end{aligned}$$

are all true with probability at least $(1-\delta)$.

Let w' be such that $\frac{R}{3} \leq \frac{F_0}{2^{w'}} \leq \frac{2R}{3}$, thus the expected number of k-mers that hash to level w' is $\frac{F_0}{2^{w'}}$ as close to $\frac{R}{2}$ as possible. The observed number of k-mers mapping to this level is a binomial random variable and can be bounded by a simple Chernoff bound

$$\begin{aligned}
\Pr\left[|N_{w'} - \frac{F_0}{2^{w'}}| > \frac{R}{12}\right] &\leq 2\exp\left(-\frac{\left(\frac{R/2}{\frac{F_0}{2^{w'}}}\right)^2}{3}\frac{F_0}{2^{w'}}\right)\\
&= 2\exp\left(-\frac{R}{18}\frac{R}{\frac{F_0}{2^{w'}}}\right)\\
&\leq 2\exp\left(-\frac{R}{18}\cdot\frac{3}{2}\right)\\
&\leq \frac{\delta}{4}
\end{aligned}$$

Thus w' is the level chosen by our algorithm and $\frac{R}{4} \leq N_{w'} \leq \frac{3R}{4}$. This implies a lower bound on the probability p_0

$$p_0 = \left(1 - \frac{1}{R}\right)^{N_{w^*}} \geq \left(1 - \frac{1}{R}\right)^{\frac{3R}{4}} \geq \frac{1}{3}$$

Note that that p_0, p_1 and x_1 are all based on counting number of k-mers, and are deterministic conditional on the hash values of all k-mers. Changing the hash value of one k-mer can affect x_1 by at most 1, and p_0, p_1 by at most $\frac{1}{R}$. Since p_0 and p_1 are functions of N_{w^*} hash values and $\frac{R}{4} \leq N_{w^*} \leq \frac{3R}{4}$, this implies that $p_0 \geq \frac{1}{3}$. Thus using the Azuma-Höffding inequality we get

$$\begin{aligned}
\Pr\left[|p_0 - \hat{p_0}| > \frac{\varepsilon}{3}p_0\right] &\leq 2\cdot\exp\left(-\frac{(\frac{\varepsilon}{3}p_0)^2}{2N_{w^*}\frac{1}{R^2}}\right)\\
&\leq 2\cdot\exp\left(-\frac{\varepsilon^2 R\cdot p_0^2}{72}\right)\\
&= 2\cdot\exp\left(-\lambda^2\log(\frac{8}{\delta})\right) \leq \frac{\delta}{4}
\end{aligned}$$

Given that there are f_1 singleton k-mers and each has a probability $\frac{1}{2^{w^*}}$ of hashing to level w^* we get, using a regular Chernoff bound

$$\begin{aligned}
\Pr\left[|2^{w^*}x_1 - f_1| > \frac{\varepsilon}{3}f_1\right] &\leq 2\cdot\exp\left(-\frac{(\frac{\varepsilon}{3})^2}{3}\frac{f_1}{2^{w^*}}\right) \\
&\leq 2\exp\left(-\frac{\varepsilon^2}{12}\frac{F_0}{\lambda 2^{w^*}}\right) \\
&\leq 2\exp\left(-\frac{\varepsilon^2}{12}\frac{R}{3\lambda}\right) \\
&\leq 2\exp\left(-\lambda\log(\frac{8}{\delta})\right) \\
&\leq \frac{\delta}{4}
\end{aligned}$$

Here we have used the inequality $f_1 \geq \frac{F_0}{\lambda}$

To show that $\Pr\left[|p_1 - \hat{p_1}| > \frac{\varepsilon}{3}p_1\right] \leq \frac{\delta}{3}$ we need a lower bound on p_1. First note

$$\begin{aligned}
x_1 &\geq (1-\varepsilon)\frac{f_1}{2^{w^*}} \\
&\geq (1-\varepsilon)\frac{F_0}{2^{w^*}}\cdot\frac{1}{\lambda} \\
&\geq (1-\varepsilon)\frac{R}{3\lambda}
\end{aligned}$$

this in turn implies that $p_1 = \frac{x_1}{R}p_0 \geq \frac{1}{10\lambda}$, assuming $\varepsilon < \frac{1}{10}$.

Using Azuma-Höffding again we get

$$\begin{aligned}
\Pr\left[|p_1 - \hat{p_1}| > \frac{\varepsilon}{3}p_1\right] &\leq 2\cdot\exp\left(-\frac{(\frac{\varepsilon}{3}p_1)^2}{2N_{w^*}\frac{1}{R^2}}\right) \\
&\leq 2\cdot\exp\left(-\frac{\varepsilon^2 R\cdot p_1^2}{72}\right) \\
&= 2\cdot\exp\left(-\log(\frac{8}{\delta})\right) \leq \frac{\delta}{4}
\end{aligned}$$

By the union bound the probability that any of the 4 bounds fail is at most δ which implies that our estimates are accurate to within a factor $1 - \frac{\varepsilon}{3}$ with probability at least $1-\delta$.

Having shown that our estimates are accurate we get,

$$\begin{aligned}\hat{f_1} &= 2^{w^*}\hat{x_1} = 2^{w^*}(R-1)\frac{\hat{p_1}}{\hat{p_0}} \\ &= (1 \pm \frac{\varepsilon}{3})^2 2^{w^*}(R-1)\frac{p_1}{p_0} \\ &= (1 \pm \frac{\varepsilon}{3})^2 w^{w^*} x_1 \\ &= (1 \pm \frac{\varepsilon}{3})^3 f_1 \\ &= (1 \pm \varepsilon) f_1\end{aligned}$$

Which shows that the estimator yields an ε-approximation with probability at least $(1-\delta)$.

□

References

[1] Michael Mitzenmacher and Salil Vadhan. Why simple hash functions work: exploiting the entropy in a data stream. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 746–755. Society for Industrial and Applied Mathematics, 2008.